# Efficient delay routing* (extended abstract)

Miriam Di Ianni[1]

Dipartimento di Scienze dell'Informazione, University of Rome "La Sapienza", via Salaria 113, I-00198 Rome, Italy. E-mail: diianni@dsi.uniroma1.it

**Abstract.** In this paper the computational complexity of finding minimum end-to-end delay packet routing schemes is studied. The existence of polynomial-time algorithms able to optimize both the end-to-end delay achievable when the number of packets in the network increases, and the number of packets that can be accepted in the network in order to keep the end-to-end delay within a constant value is investigated. In particular, it is proved the hardness of approximating in polynomial time both the minimum end-to-end delay and the maximum number of accepted packets even within a sublinear error in the number of packets.

## 1 Introduction

Efficient routing of messages is a fundamental task in parallel and distributed systems. Many packet routing algorithms trying to minimize the completion time of delivering packets have been proposed in the past [9, 11, 12, 17] but they were mainly devised for particular topologies. The first step towards the design of topology independent routing algorithms was the randomized technique proposed by Valiant and Brebner [24, 25], even if the authors first used it to route on hypercubes. Successively, a series of fundamental papers [1, 10, 14, 20, 21, 22, 23] showed the effective advantage of randomization in the design of efficient routing strategies. Universal deterministic packet routing was significantly approached by Leighton, Maggs Ranade and Rao [13, 14, 15]. Their solution to routing consists of two steps: during the first one the paths to be followed by packets are selected, while the second step, usually called *scheduling*, is used for the timing of packet movements is decided in order to minimize the total delivery time without violating network constraints (limited channel bandwidth and queue size). In [13], the authors proved the existence of a schedule bringing all packets to their respective destinations in $O(C + D)$ steps for any set $P$ of paths used to route the packets, where $C$ is the *congestion* (maximal number of paths in $P$ using the same channel) and $D$ the *dilation* (length of the longest path in $P$) of $P$. In [15] a polynomial time algorithm able to find such a schedule has been shown.

Usually, the delivery time of a packet is expressed as the sum of *network latency* and *end-to-end delay*. The first quantity is strictly dependent on the

* Work supported by the "Human Capital and Mobility" MAP project.

architectural choices for the network (topology, switching technique) and it is not affected by network congestion; instead, the end-to-end delay measures the number of times a packet must wait for traversing the links because of their limited bandwidth. In order to study the effects of network congestion on its performance, in this paper the attention is focused on the end-to-end delay: given a network and a set of packets, route them in order to minimize the maximum end-to-end delay. The interest in minimum end-to-end delay schedules, is also motivated by other reasons. The first one relies on some switching techniques yielding network latencies which can be considered almost independent from the lengths of the paths (virtual cut-through and wormhole routing). In such cases, the end-to-end delay is the main factor which strongly affects the delivery time of packets. The second reason is more theoretical and is related to the results in [13, 15]: even if the bound $O(C + D)$ on the delivery time of a schedule is asymptotically optimal, it was still unknown if a schedule that completes the delivery of packets within $D+O(C)$ steps can be found in in polynomial time. Notice that, while $D$ is a "physical" constraint on the delivery time of any schedule, the number of steps required after the first $D$ ones is a measure of the goodness of a schedule with respect to the congestion.

### 1.1 Results and paper organization

The minimum end-to-end delay problem has already been considered in [4, 5], where the authors proved the hardness of optimally routing and of approximating the minimum end-to-end delay in a network model in which the main resource packets must share is storage inside nodes. It will be called *buffer quarrel model.* In this paper, the results of [4, 5] are extended to a model in which an unbounded amount of buffers is associated to every node while the bandwith is kept bounded. This model will be called *channel quarrel model.* In this case, the main resource packets must share is edge bandwidth.

Both the routing paradigm proposed in [13], of first choosing the paths and then scheduling channel assignments, and the more general one, in which the choice of the paths and the scheduling are interleaved, are considered. The former will be called *fixed paths routing*, the latter *arbitrary paths routing.*

In section 2.1 it is proved that, in the fixed paths routing case, the minimum end-to-end delay cannot be approximated within a relative error in $O(k^{\frac{1}{13}-\delta})$ for any $\delta > 0$, where $k$ is the number of packets, unless P=NP. It follows that it is impossible to find in polynomial time schedules whose delivery time is $C+O(D)$. Such result can be easily extended (via generalization) to the arbitrary paths routing case.

Assuming knowledge of the whole network available at each node is somehow unrealistic in large distributed systems since this requires a great amount of information to be exchanged between nodes. A relevant issue is thus to investigate the performance of *local strategies*, i.e. routing algorithms in which nodes send packets according to the knowledge of their neighbors' state only. A first contribution to this aim has been given in [5] by introducing a "local optimum" criterium: if two or more packets simultaneously require a buffer in a node and one of them can alternatively choose another node belonging to a different shortest path towards its destination and which is not requested by any other packet then it will use such node. In fact, if each node has the knowledge of the occupancy state of its neighbors only, it has not sufficient information for delaying a packet which can advance along another shortest path. The behavior of similar greedy strategies has already been investigated in [3, 19] with respect to the minimum delivery time and, as remarked in the second paper, such policies are widely used. While the hardness result for fixed paths routing can be easily extended to local strategies, a stronger result has been proved for arbitrary paths routing. In this case, approximating the minimum end-to-end delay with respect to local strategies within a relative error of $f(k)$ is NP-hard for *any* sublinear function $f$ in the number $k$ of packets. In section 2.2, the idea of "local greedy schedule" is considered in the channel quarrel model and the above result is extended to it.

Next, a flow control mechanism is considered: a call admission algorithm selects a subset of communication requests which can be satisfied within a fixed end-to-end delay. In section 3 it is proved that it is NP-hard to approximate the maximum number of packets which can be accepted in the network in order to schedule them with no end-to-end delay within a relative error in $O(r^{\frac{1}{8}-\delta})$, for any $\delta > 0$, where $r$ is the number of communication requests.

In the last section conclusive remarks and open questions are discussed.

## 1.2 Preliminary definitions

A network is usually represented as a graph, where nodes stand for sites containing the processing elements and edges for communication links. A *bandwidth* is associated to each edge, representing the maximum number of packets which can be simultaneously transmitted on it. Each packet in the network follows a route starting at its *source node* and ending at its *destination*, and each transmission along one edge requires one unit of its bandwidth. If a packet cannot be transmitted along one link at a given time, it is stored in a buffer included in the output queue of that link. In this paper, it is always assumed that edge bandwidth is 1 and that transmission along one edge takes one time unit.

A network is in the *initial configuration* if all the packets are into their respective source nodes; it is in the *final configuration* if each packet has reached its own destination. The schedule ends when the network reaches the final configuration and the number of configurations met by a schedule $S$ to end is called *delivery time* or *length* of $S$. If in a configuration $i$ met by a schedule $S$ a packet $p$ has neither reached its destination nor is transmitted along any edge, then $p$ is said to be *delayed* at $i$ by $S$. The *delay* $d(p, S)$ of packet $p$ denotes how many times $p$ has been delayed by $S$. The end-to-end delay $d(S)$ of schedule $S$ is the maximum of the $d(p, S)$'s.

Since the main results presented in this paper are hardness proofs, and since if a problem is intractable even under particular conditions then, a fortiori, it is intractable in the general case, some restrictions are imposed to the model in order to make the results stronger. Thius, unless differently stated, *layered* *networks* are always considered: nodes are partitioned in $L+1 \geq 2$ sets or *levels* $V^i$, with $0 \leq i \leq L$ and edges exist only between consecutive levels. Furthermore, source nodes are always included in level 0, and destinations are always included in level $L$. Finally, only off-line schedules are considered in this paper, that is, all the packets are known before the schedule is started.

## 2 Approximating the minimum end-to-end delay

The attention in this section is focused on polynomial time algorithms able to find approximate solutions, that is, solutions whose sizes have bounded relative error with respect to sizes of the optimal ones. Here, the relative error of an algorithm $A$ for a minimization problem $\Pi$ is defined as follows:

$$\frac{m(S_A(x))}{m(S^*(x))}$$

where $S^*(x)$ is an optimum solution relative for instance $x$ of $\Pi$, $S_A(x)$ is the approximate solution found by $A$, and $m(S^*(x))$ and $m(S_A(x))$ are, respectively, their sizes. A problem is said to be $\epsilon$-approximable if a polynomial time algorithm $A$ exists such that the relative error is never greater than $\epsilon$.

In [4, 5] the minimum delivery time schedule and the minimum end-to-end delay schedule problems have been studied in the buffer quarrel model in which a packet cannot be transmitted from node $u$ to node $v$ if $v$ does not contain a free buffer when the transmission is requested. Although the two models look quite similar, it does not seem easy to transform a buffer quarrel network $N$ into a channel quarrel network $N'$ in such a way that there is a strong relation between any schedule in $N$ and the corresponding schedule in $N'$.

In fact, an intuitive transformation of $N$ into $N'$, would map each node $u$ of $N$ into an edge $(u, u')$ of $N'$ and assign to it a bandwidth equal to the number of buffers included in $u$. However, this transformation is not correct. Consider for instance the buffer quarrel network $N$ in figure 1(a) in which every node contains one buffer: by making packet $p_2$ occupy earlier than $p_1$ the buffer contained in $u$ and packet $p_3$ occupy earlier than $p_2$ the buffer contained in $v$, we get a schedule of length $L+4=7$ in which no pair of packets reach their destinations at the same time. In figure 1(b), the network $N'$ corresponding to $N$ (according to the previous described transformation) is shown: in this case edges $(u, u')$ and $(v, v')$ have both bandwidth 1. By using the same priorities as before in order to assign edges to packets (i.e., edge $(u, u')$ is assigned first to $p_2$ and edge $(v, v')$ is assigned first to $p_3$), we get a schedule of length $L+1$ in which packets $p_1$ and $p_2$ reach their destinations at the same time.

In spite of the previously remarked differencies, in the model considered throughout this paper it is still possible to prove very similar results to the ones proved in [4, 5] for the buffer quarrel model.

### 2.1 Centralized strategies

In this section strategies that take decisions about packet transmissions according to the knowledge of the state of the entire network are considered.

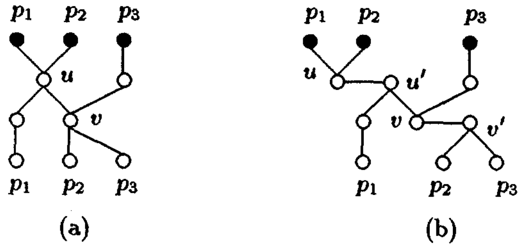


**Fig. 1.** Transformation of a buffer quarrel network into a channel quarrel network.

**Theorem 1.** *In the fixed paths routing case, the minimum end-to-end delay problem cannot be approximated with an error in* $O(k^{\frac{1}{13}-\delta})$ *for any* $\delta > 0$, *where* $k$ *is the number of packets to be transmitted, unless* P=NP.

*Proof.* The proof is similar to the one presented in [5]. Let us consider the well-known NP-hard min-colorability problem [6]: given a graph $G = (V, E)$, find the minimum-size partition $V_1, \ldots, V_{h_{min}}$ of $V$ such that no pair of nodes contained in a same $V_i$ are adjacent in $G$. A reduction from min-colorability is shown such that the original graph $G$ can be colored with $h$ colors if and only if the corresponding network admits a schedule having end-to-end delay $h - 1$.

Let $\langle G = (V, E)\rangle$ be an instance of min-colorability with $n = |V|$ and $m = |E|$. The reduction maps $\langle G = (V, E)\rangle$ into a network $N^G$ containing $n$ source nodes $s_i$, each of them corresponding to a node of $G$. The packet contained in $s_i$ will be denoted as $x_i$.

Basically the network is a chain of $n$ identical *filters* such that the $n$ outputs of one filter are connected to the $n$ inputs of the next one (see figure 2 (c)). The goal of a filter is to create conflicts between pairs of packets representing adjacent edges in the input graph $G$. Each filter is a layered network with $n$ source and $n$ destinations. The pair of inner levels $(2i - 1, 2i)$ corresponds to edge $e_i = (v, w) \in E$ of the input graph $G$: each of them contains $n - 1$ nodes with one node in level $2i - 1$ having indegree two and one node in level $2i$ having outegree two. Such nodes are connected by an edge which must be used by both of the two packets representing $v$ and $w$. All the others nodes have indegree and outdegree one. Thus, there are $2m$ inner levels (see figure 2 (b)). Clearly, such a network can be constructed in polynomial time.

To prove that $G$ can be colored with $h \leq n$ colors if and only if a schedule for the network exists with end-to-end delay $k = h - 1$ suppose first that a partition of $V$ into $h$ subsets $V_1, \ldots, V_h$ having the required property exists. In this case, packets are partitioned into $h$ sets $S_1, \ldots, S_h$ where each $S_i$ contains packets associated to the nodes in $V_i$: packets in $S_i$ leave level 0 (i.e. they start) with a delay equal to $i - 1$, for $i = 1, \ldots, h$. Since nodes in $V_i$ are pairwise not adjacent, all packets included in a same set $S_i$, $i = 1, \ldots, h$, use distinct edges within the network and, thus, they are not furtherly delayed. Hence, all packets in $S_i$ reach their destinations with delay $i$, that is, the end-to-end delay of such a schedule is $h - 1$.

Conversely, suppose the network admits a schedule with end-to-end delay $h - 1$ ($h \leq n$). Hence, the packets arrive at the end of the last filter partitioned into $h$ sets $S_1, \ldots, S_h$. Thus, none of the $S_i$ contains any pair of packets which are in conflict for the use of some edge. Indeed, this property is true when the set of packets having delay 0 (i.e. the set $S_1$) leaves the first filter. However, some pair of packets leaving the first filter with the same delay may have a conflict for the use of some edge in the

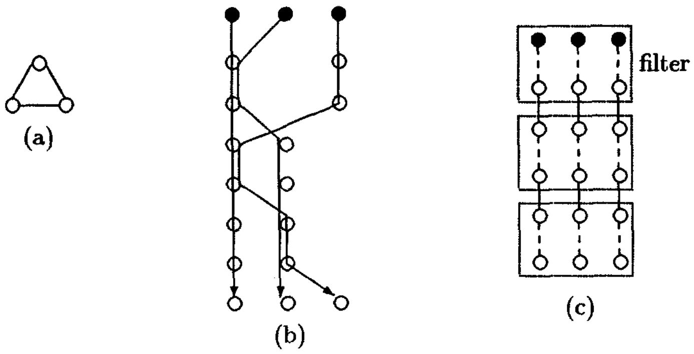


**Fig. 2.** An example of the reduction: (a) graph $G$, (b) filter, (c) the network as a chain of filters.

next levels: for instance, this happens if packet $x$ and packet $y$ have a conflict in level $i$ of the first filter and both of them are delayed by some packet in $S_1$, respectively, in levels $j_x > i$ and $j_y > i$. But, at the end of the second filter, packets having delay 1 (i.e. the set $S_2$) are pairwise non conflicting. In general, at the end of the $i$-th filter this property is true for the set $S_i$. Thus, after at most $h$ filters none of the sets $S_i$ can contain conflicting packets. It follows that nodes corresponding to packets included in a same set $S_i$ are pairwise not adjacent, that is, nodes of $G$ can be partitioned into $h$ pairwise disjoint subsets $V_1, V_2, \ldots, V_h$, where each $V_i$ contains the nodes corresponding to packets included in $S_i$.

Since the delay of the schedule and the size of the coloring are linearly related, the previous reduction preserves approximation properties. Indeed, suppose that a polynomial-time $\epsilon(k)$-approximation algorithm $A$ for the minimum end-to-end delay problem in the fixed paths routing case exists ($k$ being the number of packets to be transmitted), that is, there exists a $\epsilon(k)$ such that for any networks $N$, $A$ yields a scheduling $S(A, N)$ whose end-to-end delay $d(S(A, N))$ satisfies the following relation:

$$\frac{d(S(A,N))}{d^*(N)} < \epsilon(k)$$

where $d^*(N)$ denotes the optimum end-to-end delay for the network $N$. Then, $A$ can be applied to the network $N^G$ corresponding to some graph $G$ with $k$ nodes according to the reduction above: consider the partition $V_1, \ldots, V_{h(A)}$ for the nodes of graph $G$ induced by the scheduling $S(A, N^G)$. Since the size $h_{min}$ of an optimum coloring for $G$ is $h_{min} = d^*(N) + 1$ then

$$\frac{h(A)}{h_{min}} = \frac{d(S(A,N)) + 1}{d^*(N) + 1} < \epsilon(k) + 1$$

This implies that any $\epsilon(k)$-approximation algorithm for the minimum end-to-end delay problem can be used also for the min-colorability one with the same performances. Since the min-colorability problem cannot be approximated with an error in $O(k^{\frac{1}{13}-\delta})$ for any $\delta > 0$ [18, 2], the assertion is proved.

The previous theorem can be easily extended to the arbitrary paths routing case by noticing that arbitrary paths routing can be viewed as a generalization of fixed paths routing, in which each entry of the routing table consists of one outgoing edge only.

Since the length $l$ of a schedule $S$ for an $L$+1-levels layered network $N$ satisfies the relation $l = L + d(S)$, theorem 1 implies that it is impossible to optimally solve in polynomial time the minimum length schedule problem, unless P=NP. However, the same cannot be said concerning its approximation properties.

## 2.2 Local strategies

In large distributed systems centralized strategies are somehow unrealistic, since they require a great amount of information to be exchanged between nodes in the network. More frequently, the strategy to solve collisions and to decide the outgoing edge along which to forward a packet at a given time is chosen by each node according to the knowledge of the state of its neighborhood only. A relevant issue is thus to investigate the performance of such *local strategies.* As a first step towards this aim, it is assumed that if two or more packets are simultaneously requiring a transmission along the same edge and one of them can alternatively choose another edge belonging to a shortest path towards its destination and which is not requested by any other packet, then it is forced to advance along that edge. This is a sort of "local optimum" criterium: if each node has the knowledge of the occupancy state of its incident edges only then it has not sufficient information for delaying a packet which can advance along another shortest path. Due to its similarity with deflection routing, this kind of schedules will be called *deflection schedules.*

Observe that, according to the standard definition of approximability [6, 7], the performance of a local strategy should be compared to the performance of an optimal one, no matter if this optimum can be actually achieved by a local strategy. This clearly implies that all negative results obtained for global strategies can be immediately extended to the performances achieved by any local strategy. However, this criterium is often too pessimistic, that is, if global strategies are unrealistic, the performance of a local strategy should be compared to the optimum achievable by local strategies (see [8] for more discussions). In what follows, the last criterion will be adopted.

In the fixed paths routing case, just notice that, in the proof of theorem 1, the optimal strategy forces packets to advance whenever it is possible. Thus, theorem 1 can be extended to local strategies and it also holds for the new criterion. A stronger result can be proved for the arbitrary paths routing case.

**Theorem 2.** *In the arbitrary paths routing case, the minimum end-to-end delay deflection schedule problem is not approximable within an error $f(k)$, where $f$ is any sublinear function and $k$ is the number of packets, unless* P=NP.

*Sketch of Proof.* The proof is based on a reduction from the disjoint connecting paths problem (in short DCP) that produces a large gap between the minimum end-to-end delay in a network corresponding to a yes instance of DCP and the minimum end-to-end delay in a network corresponding to a no instance. The DCP problem is defined as follows: given a graph $G$ and a set $\{(s_1, t_1), (s_2, t_2), \ldots (s_h, t_h)\}$ of pairs of nodes of $G$, decide if $G$ contains $h$ pairwise disjoint paths, each connecting a pair $(s_i, t_i)$, $i = 1, \ldots, h$. DCP is a well-known NP-complete problem [6] and it has been recently proved to remain NP-complete also for instances restricted to layered graphs [4].

Given graph $G$ with $L_G$ levels and $h$ pairs of nodes, the network $N^G$ of the corresponding instance of the minimum end-to-end delay deflection schedule problem is composed by $h$ identical subnetworks $N^1, N^2, \ldots, N^h$ plus a final 'funnel' $F$. Each $N^i$ contains $2L_G + 2$ levels and is partitioned into two further subnetworks, $N_1^i$ and $N_2^i$. $N_1^i$ contains $h$ pairs of source nodes and the two packets belonging to a pair, $x_j^i$ and $y_j^i$, are forced to use the same edge. This device is used in order to avoid 0 delays that are inconsistent with the definition of relative error. The pair of levels 2 and 3 of $N_1^i$ corresponds to the first two levels of $G$ and, in general, the pair of levels $2j$ and $2j+1$ of $N_1^i$ corresponds to the pair of levels $j$ and $j+1$ of $G$. The remaining pairs of levels ($2j-1$ and $2j$) only contain 'vertical' edges, that is, edges connecting pairs of nodes of $N_1^i$ that correspond to the same node in $G$. $N_2^i$ does not contain any source node and starts at level 4; furthermore, each level of $N_2^i$ contains $h$ nodes and $N_2^i$ consists of a set of $h$ disjoint chains. Finally, level $2j+1$ of $N_1^i$ and level $2j+2$ of $N_2^i$, $j \geq 1$, are a complete bipartite graph. $F$ contains 3 levels and the first and the second levels contain $h^2+1$ nodes: all nodes of the last level of $N_2^i$, $i = 1, \ldots, h$, are connected with the last node of the first level of $F$, while node $j$ of the last level of $N_1^i$ is connected with node $h(i-1)+j$ of the first level of $F$. Node $j$ of the first level of $F$ is connected with node $j$ of its second level which, in turn, is connected with the $j$th destination, $j = 1, 2, \ldots, h^2$. Finally, the last node of the first level of $F$ is connected with the last node of second level which, in turn, is connected with every destination.

In figure 3 it is shown an example of the reduction. For the sake of simplicity, levels 0 and 1 of $N_1^1$ and $N_1^3$ have not been drawn and only packets $x$ have been depicted; finally, the last node of the first two levels of $F$ has been drawn in the center.

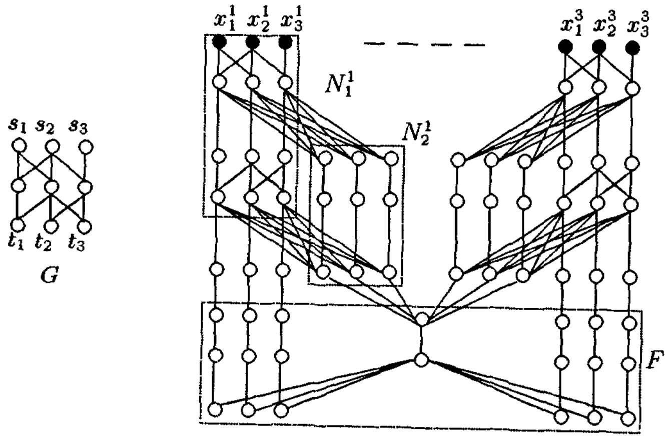


**Fig. 3.** Graph $G$ and the corresponding $N^G$.

Notice that, if a packet $x_j^i$ or $y_j^i$ passes through $N_2^i$ then there is a unique (shortest) path it can follow to reach its destination. Notice also that $N_1^i$ contains $h$ disjoint paths between the source and the destinations of each packet $x_j^i$ (or $y_j^i$) if and only if the input graph $G$ contains $h$ disjoint paths between the pairs $(s_j, t_j)$ of nodes.

Thus, if $G$ contains $h$ disjoint paths between the pairs $(s_j, t_j)$ an optimum deflection schedule with delay $d^* = 1$ is easily achieved when all the $x_j^i$'s follow the disjoint paths in $N_1^i$, followed by the $y_j^i$'s in a pipeline fashion, for any $i = 1, \ldots, h$. Conversely, if $G$ does not contain the $h$ disjoint paths, all deflection schedules have end-to-end delay $d \geq 2h - 1$. In fact, whenever two packets $x_{j_1}^i$, $x_{j_2}^i$ (or $y_{j_1}^i$, $y_{j_2}^i$) have a conflict for using some edge of $N_1^i$, one of them is forced to advance in the first node of the next level of $N_2^i$, according to the definition of deflection schedule. This implies that all of them have to pass through the 'funnel' in the last node of the first level of $F$.

Suppose now an $f(h)$-approximation algorithm $A$ for the minimum end-to-end delay schedule problem exists, where $f(h)$ =o$(h)$. Without loss of generality, since $f(h)$ =o$(h)$, consider an instance of DCP in which $h > \frac{f(h)+1}{2}$: apply to it the reduction above and, finally, apply $A$ to the corresponding $N^G$. If the $h$ disjoint paths exist, $A$ finds a deflection schedule having end-to-end delay $d \leq f(h)$. Similarly, if the $h$ disjoint paths do not exist, the algorithm finds a deflection schedule having end-to-end delay $d \geq 2h - 1 > f(h)$. This implies that the $f(h)$-approximation algorithm for the minimum end-to-end delay deflection schedule problem corresponds to a polynomial-time algorithm which decide the DCP problem, an absurd.

Since $k = 2h^2$, it has been proved that the minimum end-to-end delay deflection schedule problem cannot be approximated with an error in $O(k^{\frac{1}{2}-\delta})$ for any $\delta > 0$. Notice now that the same reasoning can be repeated for a network $N^G$ composed by $h^r$ identical subnetworks $N^1, N^2, \ldots, N^{h^r}$, for any $r > 0$: in this case, it can be shown that the minimum end-to-end delay deflection schedule problem cannot be approximated with an error in $O(k^{\frac{r}{r+1}-\delta})$ for any $\delta > 0$. Since this claim holds for any $r > 0$, the assertion is completely proved.

It is interesting to observe that the network $N^G$ in the proof of the previous theorem can be easily modified in order to place the funnel $F$ arbitrarily 'far' from the position in which the scheduling decisions that generate a 'big' delay are taken. In other words, a more 'efficient' approximating schedule could be derived only if every node can keep the knowledge of the entire network. Only in this case, a node would be able to correctly decide to delay a packet rather than sending it to a 'free' shortest path.

## 3 Call admission

In this section the attention is focused on call admission flow control mechanisms: whenever a set of nodes in the network wants to send messages to other nodes, the call admission algorithm selects a subset of the requests which can be satisfied within a fixed end-to-end delay. Next theorem proves that maximizing or even approximating the number of admitted requests in polynomial time is an hard problem also when the required fixed end-to-end delay is 0. Here, the relative error of an algorithm $A$ for a maximization problem $\Pi$ is defined as follows:

$$\frac{m(S^*(x))}{m(S_A(x))}$$

where $S^*(x)$, $S_A(x)$, $m(S^*(x))$ and $m(S_A(x))$ are defined similarly to section 2.

**Theorem 3.** *The maximum number of communication requests which can be accepted by the system in order to be satisfied with no end-to-end delay cannot be approximated with an error in* $O(r^{\frac{1}{6}-\delta})$ *for any* $\delta > 0$, *where* $r$ *is the number of communication requests, unless* P=NP.

*Sketch of Proof.* The theorem is proved for the paths beeing fixed before the requests are submitted to the system. The assertion for the general case follows by generalization (similarly to what noticed at the end of theorem 1).

The proof is an approximation preserving reduction from the well-known NP-hard max-clique problem [6]: given a graph $G = (V, E)$ with $|V| = r$, find the maximum-size complete subgraph of $G$.

Let $\langle G = (V, E)\rangle$ be an instance of max-clique with $r = |V|$ and $m = |E|$. The reduction maps $\langle G = (V, E)\rangle$ into a layered network $N^G$ containing $r$ level 0 nodes $s_i$, each of them corresponding to a node of $G$, and $r$ level $L$ nodes $t_i$. Every pair $(s_i, t_i)$ is a communication request submitted to the flow control procedures of the system.

$N^G$ contains $L - 1 = r(r-1) - 2m$ inner levels; it is described in terms of the paths chosen for the communication requests. Each pair of inner levels $(2i-1, 2i)$ corresponds to a pair of non adjacent nodes $v$, $w$ of the input graph $G$: they contain $r - 1$ nodes with exactly one node in level $2i - 1$ having indegree two and one node in level $2i$ having outegree two. Such nodes are connected by an edge that must be used by both packets representing $v$ and $w$. The other nodes have indegree and outdegree one (see figure 4). Clearly, $N^G$ can be constructed in polynomial time.

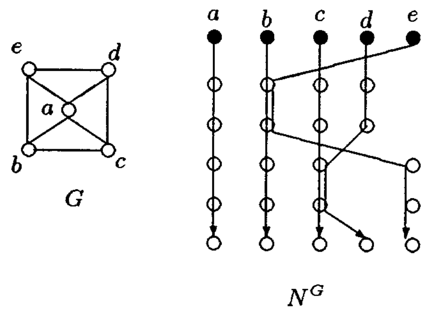


**Fig. 4.** Network corresponding to graph $G$

Let us now prove that $G$ contains a clique of $k$ nodes if and only if $k$ packets exist which can be scheduled with end-to-end delay 0. Notice that, by the above construction, a set of packets reaches level $L$ without any end-to-end delay if and only if they correspond to pairwise adjacent nodes in $G$. Thus, the above statement is trivially true.

Suppose now that a polynomial-time $\epsilon(r)$-approximation algorithm $A$ for the maximum call admission problem exists ($r$ being the number of communication requests), that is, there exists a $\epsilon(r)$ such that for any networks $N$, $A$ accepts $R(A, N)$ requests that can be scheduled with end-to-end delay 0, with

$$\frac{R^*(N)}{R(A,N)} < \epsilon(r)$$

where $R^*(N)$ denotes the maximum number of requests that can be scheduled with end-to-end delay 0. Then, $A$ can be applied to the network $N^G$ which corresponds to some graph $G$ with $r$ nodes according to the reduction described above: since the size of a maximum clique in $G$ is $k_{max} = R^*(N)$, then the above relation bounds also the ratio between the approximate and the maximum clique in $G$. In other words, $A$ can be easily transformed into an $\epsilon(r)$-approximation algorithm for the maximum clique problem. Since this last problem cannot be approximated with an error in $O(r^{\frac{1}{6}-\delta})$, for any $\delta > 0$ [2], then the assertion is proved.

# 4 Conclusions

In this paper some computational complexity results have been shown related to the problem of finding minimum end-to-end delay packet routing schemes, trying to optimize both the end-to-end delay, when the number of packets (and, thus, the congestion) increases, and the number of packets which can be accepted in the network in order to keep the end-to-end delay low. Unfortunately, all the results are negative, even with respect to approximate solutions. This means that it is impossible to design an algorithm that is efficient (with respect to running time) and performs well (with respect to the quality of solutions) in the worst case. It is thus worth to perform an average case analysis, that is, to study if the minimum end-to-end delay schedule problem is NP-complete on the average with respect to some reasonable probability distribution in the set of instances, or even to search for heuristics that are on the average approximating.

Finally, since little is known about the actual performances of the various proposed heuristics, another interesting issue would be their effective implementation and simulation in distributed environments.

# References


1. R. Aleliunas, "Randomized parallel communication", *Proc. of the ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing*, 60-72, 1982.
2. M. Bellare, M. Sudan, "Improved non-approximability results", *Proc. of the 26th Annual ACM Symposium on Theory of Computing*, 184-193, 1994.
3. I. Cidon, S. Kutten, Y. Mansour, D. Peleg, "Greedy packet scheduling", *Proc. of 4th Workshop on Distributed Algorithms*, 1990.
4. A. Clementi, M. Di Ianni, "Optimum Schedule Problem in Store and Forward networks", *Proc. of the IEEE INFOCOM'94*, 1336-1343, 1994.
5. A. Clementi, M. Di Ianni, "On the hardness of approximating optimum schedule problems in Store and Forward networks", to appear in *IEEE-ACM Transactions on Networking*, 1995.
6. M.R. Garey, D.S. Johnson, *Computers and intractability: a guide to the theory of NP-completeness*, Freeman, San Francisco, 1979.
7. D. S. Johnson, "Approximation algorithms for combinatorial problems", *Journal of Computer and System Sciences*, 9, 256-278, 1974.
8. E. Koutsoupias, C. H. Papadimitriou, "Beyond competitive analysis (on-line algorithms)", *Proc.of the 35th IEEE Annual Symposium on Foundations of Computer Science*, 394-400, 1994.

9. D. Krizanc, S. Rajasekaran and T. Tsantilis, "Optimal routing algorithms for mesh-connected processor arrays", *Lecture Notes in Computer Science*, Springer-Verlag, New York, vol. 319, 411-422, 1988.
10. T. Leighton, *Introduction to parallel algorithms and architectures: arrays, trees, hypercubes*, Morgan Kaufmann Publishers (San Mateo, Ca), 1992.
11. T. Leighton and B. Maggs, "Fast algorithms for routing around faults in multibutterflies and randomly-wired splitter networks", *IEEE Trans. on Comput.*, 41, N. 5, 578-587, 1992.
12. T. Leighton, F. Makedon and I. Tollis, "A $2N-2$ step algorithm for routing in an $N \times N$ mesh", *Proc. of the 1989 ACM Symposium on Parallel Algorithms and Architectures*, 1989.
13. T. Leighton, B. Maggs, S. Rao, "Universal packet routing algorithms", *Proc. of the 29th IEEE Annual Symposium on Foundations of Computer Science*, 256-269, 1988.
14. T. Leighton, B. Maggs, S. Rao, A.G Ranade "Randomized routing and sorting on fixed-connection networks", *Journal of Algorithms*, 17, 157-205, 1994.
15. T. Leighton, B. Maggs, "Fast algorithms for finding O(congestion + dilation) packet routing schedules", Unpublished Manuscript, 1994.
16. T. Leighton, B. Maggs, S. Rao, "Packet routing and job-shop scheduling in O(congestion + dilation) steps", *Combinatorica*, 2, 167-186, 1994.
17. C.E. Leiserson and B. Maggs, "Communication-efficient parallel graph algorithms for distributed random-access machines", *Algorithmica* 3, 53-77, 1988.
18. C. Lund, M. Yannakakis, "On the hardness of approximating minimization problems", *Proc. of 25th Annual ACM Symposium on Theory of Computing*, pp. 286-293, 1993.
19. Y. Mansour, B. Patt-Shamir, "Greedy packet scheduling on shortest paths", *Proc. of the ACM PODC '91*, 165-175, 1991.
20. F. Meyer auf der Heide, B. Vöcking, "A packet routing protocol for arbitrary networks", *Proc. of the 12th Symposium on Theoretical Aspects of Computer Sience*, 291-302, 1995.
21. N. Pippenger, "Parallel communication with limited buffers", *Proc. of the 25th IEEE Annual Symposium on Foundations of Computer Science*, 127-136, 1984.
22. A.G. Ranade, "How to emulate shared memory", *Journal of Computer and System Sciences*, 42, 307-326, 1991.
23. E. Upfal, "Efficient schemes for parallel communication", *Proc. of the ACM SIGACT-SIGOPS 1989 ACM Symposium on Principles of Distributed Computing*, 55-59, 1982.
24. L.G. Valiant, "A schemes for fast parallel communication", *SIAM Journal on Computing*, 11/2, 350-361, 1982.
25. L.G. Valiant and G.J. Brebner, "Universal schemes for parallel communication", *Proc. 13th ACM Symposium on Theory of Computing*, 263-277, 1981